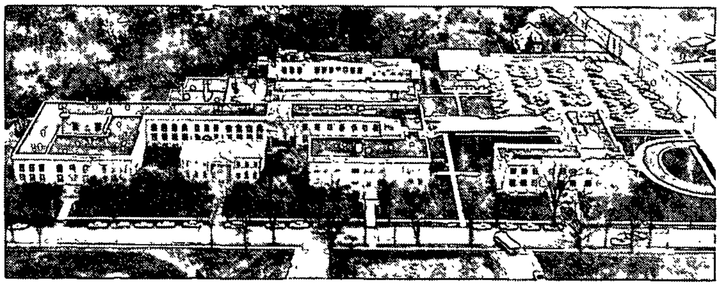


THE INSTITUTE OF PAPER CHEMISTRY, APPLETON, WISCONSIN

IPC TECHNICAL PAPER SERIES

NUMBER 103

DYSCO, A MODULAR, DYNAMIC SIMULATOR FOR THE PULP AND PAPER INDUSTRY

PETER E. PARKER

FEBRUARY, 1981

DYSCO, A MODULAR, DYNAMIC SIMULATOR FOR THE PULP AND PAPER INDUSTRY

Peter E. Parker
Group Leader
Systems Engineering Group
Environmental Sciences Division
The Institute of Paper Chemistry
Appleton, WI 54912

ABSTRACT

Steady state, modular simulation programs (GEMS, PLONTRAN) have been widely used in the pulp and paper industry. These same techniques can be incorporated into a dynamic simulation package. With a dynamic simulator, problems involving plant start up and shutdown, process control, and emergency procedures can be studied. DYSCO, DYnamic Simulation and COntrol, has been adapted for use in the pulp and paper industry. The utility and limitations of the simulator are illustrated with examples.

INTRODUCTION

Modelling is basically the process of building a copy of a real object. In the context of this paper, a model is a mathematical or computerized representation of a real process. Since the model is a representation of the real process, there are details that are not included in the model. These "deficiencies" naturally lead to limitations of the model. In addition, deliberate omission of certain levels of detail determine the applicability of the model. No one model is correct for all uses.

Simulation packages are collections of computer programs that were developed to solve specific classes of problems. Like process models, no one simulation package is applicable to all problems. Extension of a simulation package to force it to handle problems for which it was not designed is at best a risky operation and at worst a complete failure. Like process models, no one simulation package is correct for all uses.

Process simulation packages contain the best and the worst features of both process models and simulation packages. The entire package is designed to solve a given class of problems. The models contained within the package represent assumptions and approximations to solve a particular class of process models. Attempts to significantly extend these models beyond their initial intent are convincingly easy to do, but highly dangerous. Similarly, an attempt to extend the simulator to include new classes of process models is asking for trouble.

In an attempt to classify both process models and simulators, let us build a model of the real world and use this model as a framework for our discussion. Figure 1 divides the world into eight classes, with three major groups. Our model says processes can be discrete or continuous, dynamic or steady state, deterministic or stochastic.

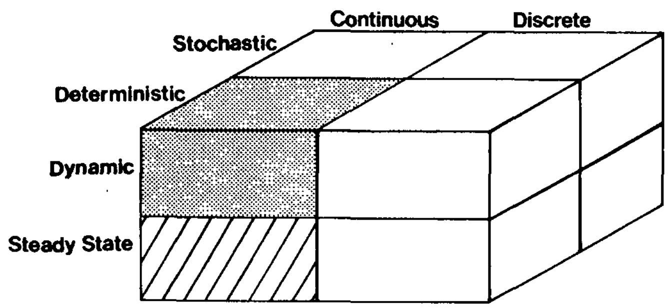


Figure 1. Possible Model Types

Discrete processes are those that operate on individual or separate items. An example of a discrete process is a mill's roll handling system. Each roll is individually weighed, wrapped, stacked, etc. There is some identification which distinguishes this roll from all others and operations are done on it as a single item - that is, each operation is discrete. Process models developed to study such systems naturally must take this individuality into consideration. Likewise, a process simulator must be capable of tracking an individual.

Continuous processes are those that utilize items that can take on a continuous range of values. Temperature is a good example. In theory, temperature can take on any value. We tend to measure temperature with a discrete number, e.g., 50°C, but in actual fact, the temperature may be an infinity of values between, say, 49.5°C and 50.5°C and still be called 50°C.

Papermaking is basically a continuous process. Furnish is withdrawn from a thick stock chest, refined, diluted, dewatered, etc., in a continuous manner. Most changes, such as basis weight, color, or furnish composition are simply step changes - the process is expected to run continuously and smoothly thereafter.

Models of continuous processes are very abundant and are generally the kind that most of us think of when we consider modelling a process. Information handling within a simulation package working with continuous models is quite different from a simulation of a discrete process.

Steady state processes are ones which do not change with time. Inputs, outputs, and processing conditions are constant. While no real process, including pulping, bleaching, or papermaking, is ever run this way, most are designed with this type of operation in mind. Control systems are used to keep the process at steady state.

Models of steady state processes can be simple or complex. Their basic function is to transform input conditions into output conditions. Steady state simulation packages are designed to efficiently solve the large set of potentially nonlinear algebraic equations that result from steady state process models.

Dynamic processes are those that vary with time. If either the feed to the production line or the process conditions vary with time, the product varies with time. All real processes are dynamic. In most cases, modelling of dynamic processes involves the solution of differential equations. Knowledge of the capacities of the various process elements becomes critical for a reasonable solution to the problem.

Dynamic process simulators must have techniques to solve the differential equations resulting from the process models. Unlike algebraic equations, there is a variety of ways to solve differential equations. The best method often depends on the structure of the system of equations and the values the various model parameters may assume. Thus, a dynamic process simulation package must have a variety of numerical differential equation solution procedures available to the user. If some of the process models must be represented by partial differential equations, sophisticated techniques must be used to efficiently obtain solutions.

Deterministic systems are those in which the process is described by a fixed set of numbers. A known input into the process produces a known output. Most of us would like to believe that a digester, for example, operates in this manner. Given a fixed wood species, alkali charge and temperature profile, a digester should produce a given amount of pulp of fixed qualtiy.

Deterministic models are the ones most often used by engineers to describe processes. The bulk of process simulators that are available are designed to work with deterministic processes.

Stochastic, or random, processes are more like the real world. Given a fixed set of inputs and process variables, the output is still a random variable. A stochastic system also results when the inputs or the process conditions vary randomly. Pulping is most probably a stochastic process. The chip feed most likely varies in quantity and quality, the temperature profile fluctuates randomly around some mean value, and the applied chemical charge also fluctuates. The output from the process is a pulp which continuously varies in quantity and quality. Statistical techniques are required to evaluate the behavior of the digester.

Simulations of stochastic processes are fundamentally different from simulations of deterministic processes. The process models, or the simulation package itself, must arrange to repeat the simulation many times with a given set of input conditions in order that statistically meaningful results are produced. The output of a stochastic simulation must be information on the mean and variance, and, possibly, higher moments, of the process.

Using these definitions, it is apparent that the production of paper is a continuous, dynamic, stochastic process. An analysis of such a process is virtually hopeless. To obtain insight into such a process, therefore, requires that the process be simplified. The first major simplification is to assume that the process is deterministic. That is, we ignore the random behavior of the model. Steady state simulation results when we ignore the dynamic behavior of the system.

Another way to classify simulation programs is by the way in which the process is described to the simulation program, and the method used to solve the resulting equations. Figure 2 illustrates this two-by-two breakdown.

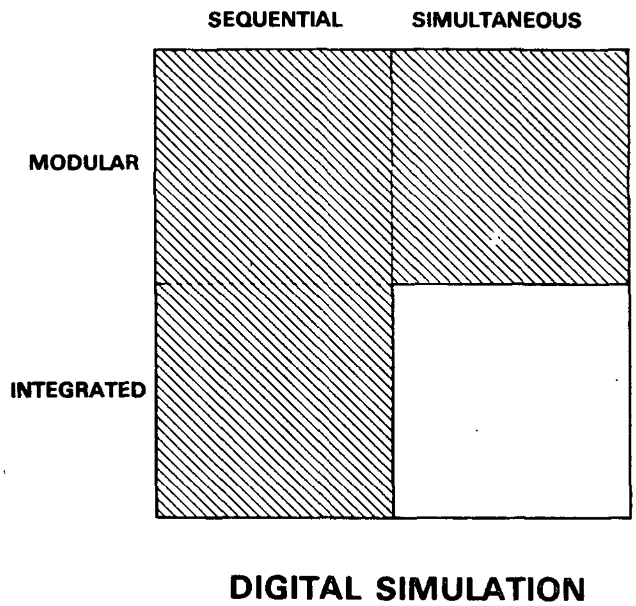


Figure 2. Categories of Digital Simulation

Modular simulators require that each particular process model be described independently of all other process models. That is, there is some structure for information transfer between process models, and each model acts independently of all other models. The major strength of this type of approach is that libraries of process models can be built and elements selected from the library when a process is to be studied. Great flexibility results from this capability. Models of varying degrees of sophistication can be utilized, depending upon the needs of the problem. Virtually all flowsheet simulators fall into this category.

Integrated simulators require that each problem be presented to it as a complete set of equations. The various inter-model relationships are explicitly stated in the equations. Information transfer between various portions of the process are direct and explicit. Analog simulation and some simulators of analog computers fall into this category. Their use is limited, as flexibility and expandability are severely limited.

Independent of how the models are presented to the simulator, the resulting equations can be solved in one of two ways. The equations can be solved simultaneously or sequentially. Algebraic equations resulting from steady state simulation can be solved simultaneously by matrix inversion techniques. Nonlinear equations and sparse matrices may require special inversion techniques, but all equations are essentially satisfied simultaneously. True simultaneous solution of differential equations would require the use of analog computers (assuming theoretical solutions are out of the question). There are ways to numerically approximate this simultaneous solution for use on digital computers, but relatively large amounts of core may be required.

Sequential solution of algebraic equations is relatively straightforward. An estimated solution is chosen for each dependent variable. These estimates are then inserted into the equations and new estimates are found. The equations are evaluated one at a time, in some predetermined order. After stepping through each equation (or process model), new estimates of the initially assumed variables are available. If the new estimates do not compare favorably to the initial estimates, a new initial estimate is made and the process is repeated. This process continues until the calculated estimate is identical, within some tolerance, to the initial estimate. If the calculated value becomes the new initial estimate, the process is called successive substitution. Convergence to the correct solution may be tedious. Various methods are available to speed up the convergence, but the basic process remains the same - new guesses are fed to the process models until calculated values agree with the initial estimate.

Sequential solution of differential equations proceeds in a somewhat similar manner to sequential solution of algebraic equations. The dynamics of each equation or process model are evaluated at a given point in time. This time derivative, plus the value of the variable(s) at some previous times, are used to determine the value of the variable(s) at the current time. If the process is sequential, each model is evaluated with the most recently available values. This tends to build a lag into the process of at least one time increment and thus this method is not often used.

DYSCO

DYSCO, DYnamic Simulation and COntrol, is a dynamic modular simulator. It attempts to solve the process in a simultaneous manner by appropriate storage of values calculated for the new point in time. In addition, it is also highly interactive, allowing the user a great deal of control over the simulation.

Two types of modules are available in DYSCO. Instantaneous models are identical to steady state models in that the outputs are algebraically related to the inputs. Dynamic models are represented by the standard mass balance equation:

$$\frac{d\,(V\cdot c)}{d\,t} = (Q\cdot c)_{in} - (Q\cdot c)_{out} + R\cdot V$$

where

V is the volume of the vessel,
c is the concentration of species,
Q is the volumetric flow rate,
R is the rate of appearance (disappearance) of the species by some conversion process,
t is time.

Each dynamic module calculates its own vector of derivatives and passes them to a numerical integrator. The integrator then returns an updated vector of concentrations.

The various modules are connected to each other by streams. The streams carry a variety of information between modules, such as pressure, temperature, flow rate, and component concentration. Information streams also exist to supply control signals to controllers and controlled equipment, such as valves.

DYSCO is a rather large program which is designed to simulate problems of various sizes. Input is processed in two distinct phases to facilitate minimization of core usage. In the first phase, the process flow sheet is entered along with some other miscellaneous information. This information is used to build two FORTRAN subroutines. One of these sets up the calls to the appropriate process models. The other sets up the dimensions of all the major arrays. Compilating these two programs and linking them to the second phase of DYSCO results in only the necessary process models being loaded from the library and dimensioning of arrays just sufficient to simulate the given problem. All of this Phase One information is entered interactively.

In Phase Two, the basic parameters describing the process are entered by the user. These data will include process model parameters and the initial condition of the system (stream flows, stream compositions, etc.). Other control information is also entered at this time. As with Phase One, this information is entered interactively.

This second phase of DYSCO conducts the actual simulation as well as processing the bulk of the process data. One of four integration methods may be selected by the user. Euler's method is most commonly used, although the variable step Adams-Moulton method is convenient. A fourth order Runge-Kutta and a fourth order, variable step Runge-Kutta are also available. The user may retain control throughout the simulation if he so desires. Whenever the user is in control, he may adjust stream or model parameters almost at will. In this manner, start up and shut down procedures may be simulated. (Automatic control of some parameters is also provided.)

PHYSICAL PROPERTIES

In most simulators, a great deal of effort has been expended to develop a good physical property data base. Such is not the case for DYSCO. When it was originally designed by Lopez[1], existing physical property data bases were intended to be used. The two data bases that were intended to be part of DYSCO were the data bases associated with UPPACER[2] and

CHESS[3]. These data bases were inappropriate for use in the pulp and paper industry and were eliminated from DYSCO. Recently, The Institute of Paper Chemistry has started to develop a physical property data base for DYSCO. Initial information includes constant heat capacities and densities. Retention information is also being included. The work of Kask[4] and Forbes[5] have demonstrated that this type of information can be used in the dynamic modelling of a paper machine. Much work needs to be done to develop this physical property data base. Use of such a data base by a dynamic simulator is inherently no different than use of such a data base by a steady state simulator, so no new problems should be expected.

APPLICATION

Two paper machine examples will be used to illustrate the utility of DYSCO. In the first example, the effect of changing retention aid dosage on the retention of fines and filler is studied. A small web former is used to supply experimental data. In the second example, the web break control system of a real, fine paper machine is analyzed.

The retention of fines and fillers is greatly affected by surface phenomena. The addition of retention aids, such as cationic polymers, change the characteristic surface phenomena and hence affect the retention of fines and fillers. While there is a strong theoretical base for predicting the qualitative effects of retention aids, experimental data are necessary to ascertain quantitative effects. Tight water system closure leads to a build-up of nonretained fiber and filler fractions. Upsets in retention aid dosage can also dramatically change the operation of a paper machine. DYSCO was used in an attempt to model a simple web former under conditions of high recyle and varying polymer dosage.

The web former system is shown in Fig. 3. Stock is metered to the system, diluted with recycled white water, and fed to a mixing vessel where retention aid is added. The stock then flows to the forming wire. The sheet is well drained due to the low speed of the forming wire before it is picked up by a couch/dryer roll. The white water collects in a wire pit and is rapidly circulated back for thick stock dilution. With the exception of the wire, modelling of this system is rather simple. Three mixer models can adequately represent the tanks and piping volumes. The wire must be represented by a separation device which includes the effect of retention aid.

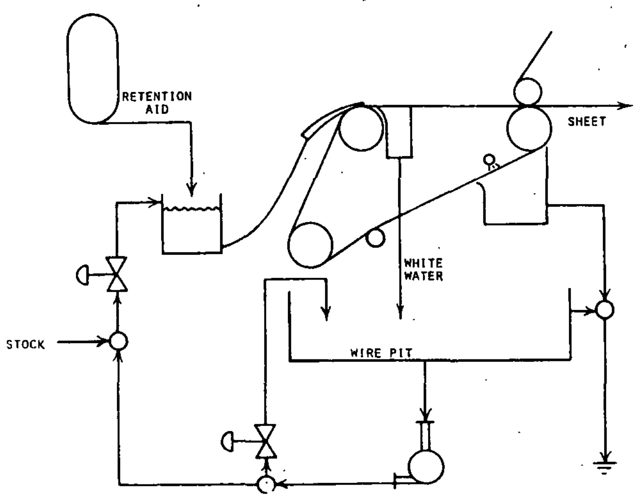


Figure 3. Experimental Web Former

The modelling of the effect of the retention aid is accomplished by incorporating some of the data of Petaja[6] into the physical property data base. Basically, Petaja performed dynamic drainage jar studies to develop retention curves similar to those in Fig. 4. Experimentally, he also determined the correct stirrer rpm which corresponded to the web former. The curves for fines and filler (TiO_2) were approximated by the straight lines also shown in Fig. 4. The constants for the straight line approximations were entered into the DYSCO physical property data base.

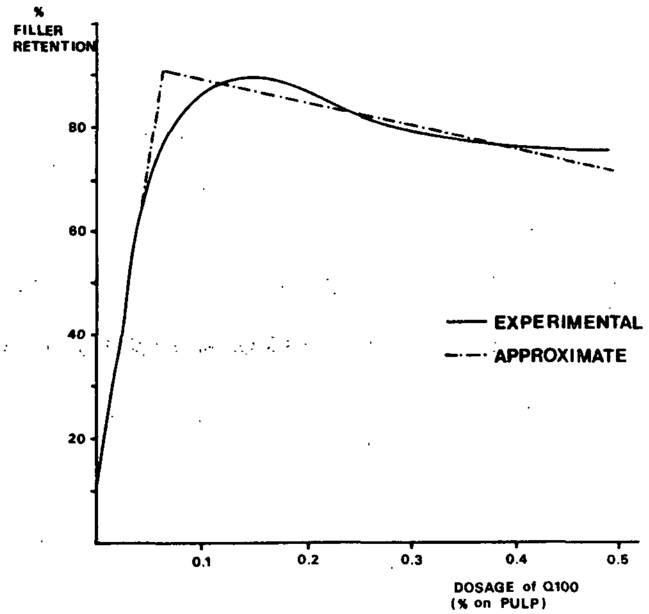


Figure 4. Effect of Retention Aid

The web former was run for about 20 minutes with a cationic retention aid added at a rate of 0.015% by weight based on pulp. The retention aid was then shut off for 19 minutes, after which it was added at the same rate.

A typical response curve is shown in Fig. 5. During the first 20 minutes, the TiO_2 in the white water approaches a steady state level of about 150 ppm. When the retention aid is shut off, less TiO_2 is retained in the sheet and the level in the white water rises. The TiO_2 levels peaks at about 270 ppm shortly after the retention aid flow is turned back on. Figure 5 shows the good agreement between the experimental data and the simulation. Similar data for fiber fines show a similar trend but do not compare quite as well with the simulation data. Overall, the agreement is quite good, considering the approximations involved in representing the retention aid curves by straight line segments.

A second major test of DYSCO was the simulation of a simplified web-break control system. A fine paper machine, shown schematically in Fig. 6, was decomposed into about 70 blocks and about 150 streams. Steady state operation was simulated and verified against experimental data.

On this paper machine, the web-break control system functions in the usual manner. When the break is sensed, the sheet is knocked into the couch pit. Broughton box water, which normally flows to the saveall, is used for the knock-down showers and additional couch pit dilution. It takes about 8 seconds to divert the full flow of this Broughton box water to the couch pit. When the couch pit consistency rises to 3.8% (from its normal 0.5%) the couch pit effluent is diverted to the broke chest from the saveall.

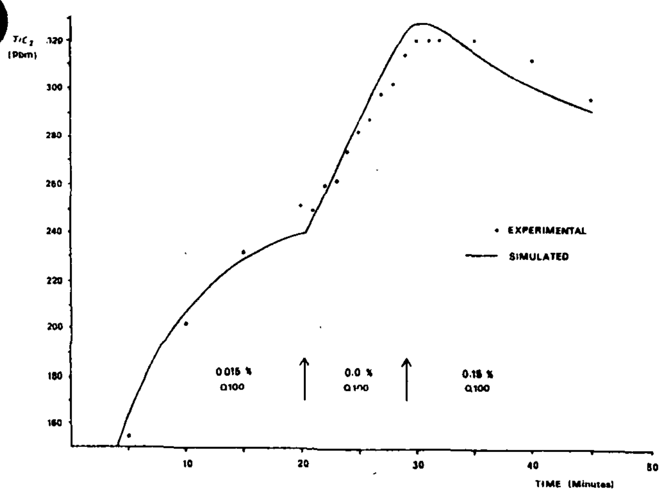
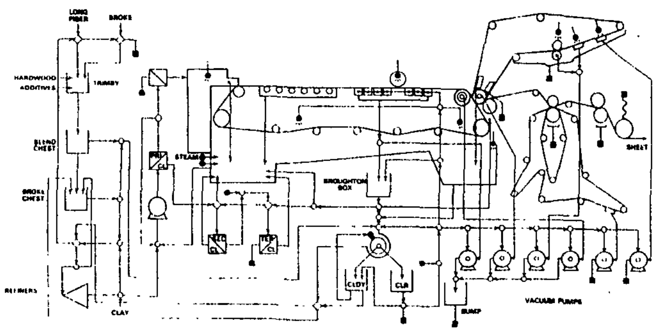


Figure 5. Response to Retention Aid Upset



PAPER MACHINE SCHEMATIC

Figure 6. Paper Machine Schematic

The control scheme causes the saveall to experience a rapidly fluctuating load. Figure 7 illustrates the simulated variation in fiber load to the saveall. Initially, the load falls when the Broughton box water is diverted to the saveall. Rather rapidly, the load rises as the diluted sheet is sent to the saveall. After about 60 seconds, the couch pit flow is diverted to the broke chest and the saveall load drops drastically. The peak load is about 30% in excess over the average, while the minimum load is about 30% of the normal load. This wide, rapid fluctuation is believed to be partly responsible for poor saveall operation during web breaks.

Engineering and control staff are looking at ways to dampen the swings.

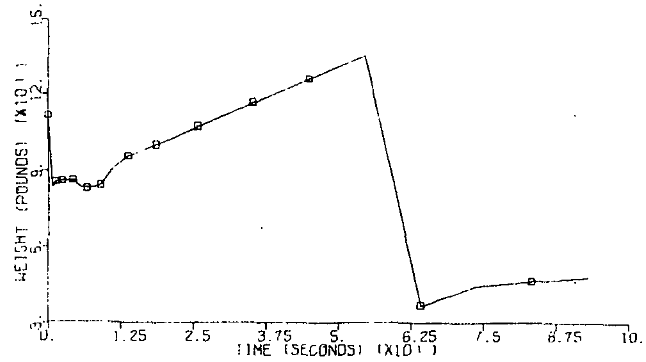


Figure 7. Fiber Load to Saveall During Web Break

CONCLUSION

DYSCO has proven itself to be a very useful simulator. The dynamic capability greatly extends the range of problems that can be studied, while its modular construction makes it relatively easy to extend its application. Student and/or novice users find that the interactive mode of operation greatly facilitates its use. The addition of a good physical property package will make DYSCO a very powerful tool for analyzing process flow sheets.

REFERENCES

1. Lopez, L., "DYSCO: An Interactive Executive Program for Dynamic Simulation and Control of Chemical Processes." Ph.D. Dissertation, Univ. of Michigan, 1974.

2. Pozuanovic, D. S., "Users Manual Physical Property Data Base Storage, Retrieval, and Estimation System." Chem. Eng. Dept., Univ. of Pennsylvania, October, 1969.

3. Motard, R. L., "Chess System Guide." Dept. of Chem. Eng. and Petro. Eng. Univ. of Houston, 1968.

4. Kask, D. W., "Dynamic Simulation of a Closed Paper Machine to Determine the Temperature of the White Water System." A-291 Report, The Institute of Paper Chemistry, 1978.

5. Forbes, M. F., "A Dynamic Simulation of Retention Aid Addition." A-291 Report, The Institute of Paper Chemistry, 1980.

6. Petaja, T., "Mechanism of Retention with Dual Polymer System." A-291 Report, The Institute of Paper Chemistry, 1979.